11

Algebraic Reconstruction Techniques

In this and the next chapter we discuss series expansion methods for image reconstruction. The *algebraic reconstruction techniques* (*ART*) form a large family of reconstruction algorithms. The name is a historical accident; there is nothing more "algebraic" about these techniques than about the techniques that are discussed in the next chapter. The distinguishing feature of ART needs careful discussion, which is given in the following section.

11.1 What Is ART?

All series expansion methods are procedures for the solution of the discrete reconstruction problem. As discussed in Section 6.3, this is the problem of estimating an image vector x such that $y = Rx + e$, given a measurement vector y. The estimation is done by requiring x, and the error vector e, to satisfy some specified optimization criterion of the type discussed in Section 6.4. We use x^* to denote the required estimate.

All ART methods of image reconstruction are iterative procedures: they produce a sequence of vectors $x^{(0)}, x^{(1)}, \ldots$ that is supposed to *converge* to x^*. This means that, for $1 \leq j \leq J$, $x_j^{(k)}$ (the jth component of the kth iterate) should be arbitrarily near to x_j^*, provided that k is chosen large enough. The process of producing $x^{(k+1)}$ from $x^{(k)}$ is referred to as an *iterative step*.

In ART, $x^{(k+1)}$ is obtained from $x^{(k)}$ by considering a single one of the I approximate equations, see (6.23). In fact, the equations are used in a *cyclic order*. We use i_k to denote $k(\text{mod } I) + 1$; i.e., $i_0 = 1$, $i_1 = 2, \ldots, i_{I-1} = I$, $i_I = 1$, $i_{I+1} = 2, \ldots$, and we use r_i to denote the J-dimensional column vector whose jth component is $r_{i,j}$. In other words, r_i is the transpose of the ith row of R. An important point here is that this specification is incomplete because it depends on how we index the lines for which the integrals are estimated. It is stated in Section 6.1 that we assume that estimates of $[\mathscr{R}f](\ell, \theta)$ are known for I pairs: $(\ell_1, \theta_1), \ldots, (\ell_I, \theta_I)$. However, until now we have not specified the geometrical locations of the lines that are parametrized by these pairs. Since the order in which we do things in ART depends on the indexing i for the set of lines for which data are collected, the specification of ART as a reconstruction algorithm is complete only if it includes the indexing method for the lines, which we refer to as the *data access ordering*. We return to this point later on in this chapter.

The kth iterative step in ART can be described by a function α_k, whose arguments are two J-dimensional vectors and one real number and whose value is a J-dimensional vector. (In mathematical jargon, $\alpha_k : \mathbb{R}^J \times \mathbb{R}^J \times \mathbb{R} \to \mathbb{R}^J$, where $\mathbb{R}$ denotes the set of real numbers.) Then,

$$x^{(k+1)} = \alpha_k \left(x^{(k)}, r_{i_k}, y_{i_k} \right). \tag{11.1}$$

In words, a particular algebraic reconstruction technique is defined by a sequence of functions $\alpha_0, \alpha_1, \alpha_2, \ldots$. In order to get the $(k+1)$st iterate we apply α_k to the kth iterate, the i_kth row of the projection matrix R, and the i_kth component of the measurement vector y. Such algorithms have been referred to as *storage efficient*, because the J-dimensional vector $x^{(k+1)}$ can be stored in the same part of computer memory where $x^{(k)}$ has been kept, since $x^{(k)}$ is not needed by the algorithm after the kth step. (Note that the implementation of the FBP described near the end of Section 8.3 is also storage efficient. The same cannot be said for the iterative procedures that are discussed in the next chapter.) Various ART methods differ from each other in the way the sequence of α_ks is chosen. We now illustrate the previous discussion on a particularly simple example.

One way of choosing the α_ks is the following. For any J-dimensional vectors x and t and for any real number z, let

$$\alpha_k (x, t, z) = \begin{cases} x + \frac{z - \langle t, x \rangle}{\langle t, t \rangle} t, & \text{if} \quad \langle t, t \rangle \neq 0, \\ x, & \text{if} \quad \langle t, t \rangle = 0, \end{cases} \tag{11.2}$$

where $\langle \bullet, \bullet \rangle$ denotes the *inner product* of two J-dimensional vectors; i.e.,

$$\langle t, x \rangle = \sum_{j=1}^{J} t_j x_j. \tag{11.3}$$

Note that, in this case, the α_ks have the same functional form for all k. Defining α_k in this way has a number of attractive properties.

One is that, if $\langle r_{i_k}, r_{i_k} \rangle \neq 0$, then

$$y_{i_k} = \sum_{j=1}^{J} r_{i_k,j} x_j^{(k+1)}, \tag{11.4}$$

i.e., the i_kth approximate equality is exactly satisfied after the kth step. To see this, combine (11.1) and (11.2) and use the notation of (11.3) to get

$$\begin{aligned}\left\langle r_{i_k}, x^{(k+1)}\right\rangle &= \left\langle r_{i_k}, \alpha_k\left(x^{(k)}, r_{i_k}, y_{i_k}\right)\right\rangle \\ &= \left\langle r_{i_k}, x^{(k)}\right\rangle + \frac{y_{i_k} - \left\langle r_{i_k}, x^{(k)}\right\rangle}{\left\langle r_{i_k}, r_{i_k}\right\rangle}\left\langle r_{i_k}, r_{i_k}\right\rangle \\ &= y_{i_k}.\end{aligned} \tag{11.5}$$

Another attractive property of defining α_k by (11.2) is that the updating of $x^{(k)}$ becomes very simple: we just add to $x^{(k)}$ a multiple of the vector r_{i_k}. In practice, this updating of $x^{(k)}$ can be computationally very inexpensive.

Consider, for example, the basis functions associated with a digitization into pixels (6.17). Then $r_{i,j}$ is just the length of intersection of the ith line with the jth pixel. This has two consequences. First, most of the components of the vector r_{i_k} are zero. At most $2\ell - 1$ pixels can be intersected by a straight line in an $\ell \times \ell$ digitization of a picture. Thus, of the ℓ^2 components of r_{i_k}, at most $2\ell - 1$ (and typically only about ℓ) are nonzero. Second, the location and size of the nonzero components of r_{i_k} can be rapidly calculated using a DDA from the geometrical location of the i_kth line relative to the $\ell \times \ell$ grid, as discussed in Section 4.6. Thus, the projection matrix R does not need to be stored in the computer. Only one row of the matrix is needed at a time, and all essential information about this row is easily calculable. For this reason such methods are also referred to as *row-action methods*.

We investigate this point further, since it is basic to the understanding of the computational efficacy of ART. Suppose that we have a list $j_1, \ldots, j_U$ of indices such that $t_j = 0$ unless j is one of the $j_1, \ldots, j_U$. Then evaluation of (11.3) requires only U multiplications, which in our application is much smaller than J, as discussed above. Similarly, $\langle t, t\rangle$ can be evaluated using U multiplications. Having evaluated $(z - \langle t, x\rangle) / \langle t, t\rangle$ using $2U$ multiplications (and one division), the updating of x can be achieved by a further U multiplications. This is because only those x_j need to be altered for which $j = j_u$ for some u, $1 \le u \le U$, and the alteration requires adding to x_j a fixed multiple of t_j. This shows that a single step of the algorithm, as described by (11.2), is very simple to implement in a computationally efficient way.

Apart from its computational efficiency, (11.2) is an intuitively reasonable way of producing $x^{(k+1)}$ from $x^{(k)}$. Suppose that, in addition to requiring the satisfaction of (11.4) after the kth iterative step, we impose the following conditions on the way the kth step should be carried out.

(i) Only those pixels that are intersected by the i_kth line should have their densities changed.
(ii) The density change of a pixel should be proportional to $y_{i_k} - \left\langle r_{i_k}, x^{(k)}\right\rangle$ (the "error" in the i_kth approximate equality prior to the kth step).
(iii) The change in the jth pixel should be proportional to $r_{i_k,j}$.

These conditions, nearly identical to the conditions for discrete backprojection stated in Section 7.3, uniquely determine how the α_ks are defined; and they lead to (11.2). The early literature on ART relied on justifying the algorithms by showing that they are derived from such reasonable conditions.

Before we get into the details of specific ART methods, two comments are in order. First, for (11.1) to specify the sequence $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ precisely, we need to select the *initial vector* $x^{(0)}$. The choice of $x^{(0)}$ is quite important in the practical behavior of these algorithms. More is said about this below. Second, if the version of the discrete reconstruction problem that is represented by the system of inequalities $Nx \leq q$ (see Section 6.4) is used, then the general description of ART given by (11.1) is not always adequate. In such a case, a slightly more complicated general framework may be required, but one that has essentially similar computational requirements. We discuss a similar situation in Section 11.3 in some detail.

11.2 Relaxation Methods for Solving Systems of Inequalities and Equalities

In this section we give the mathematical background to ART. We do this in the framework of the mathematical problem "find a vector that satisfies all of a given set of linear inequalities." In other words, we are interested in finding a J-dimensional vector x such that

$$\langle n_i, x \rangle \leq q_i, \quad \text{for } 1 \leq i \leq P, \tag{11.6}$$

where the n_i are given J-dimensional vectors and the q_i are given real numbers. Equation (11.6) is a rewrite of (6.42).

In what follows we assume that each n_i has at least one nonzero component. A physical interpretation of this assumption is that we do not make use of a measurement if none of the basis functions contributed to it. The reason for making this assumption is to avoid having to make special cases all the time when $\langle n_i, n_i \rangle = 0$, like we had to do in (11.2).

We introduce some sets of vectors N_i $(1 \leq i \leq P)$ and N. For $1 \leq i \leq P$,

$$N_i = \{x \mid \langle n_i, x \rangle \leq q_i\} \tag{11.7}$$

and

$$N = \bigcap_{i=1}^{P} N_i. \tag{11.8}$$

In words, N_i is the set of vectors that satisfies the ith of the P inequalities in (11.6), and N is the set of vectors that satisfies all P inequalities. Our aim, for now, is to find an element of N.

More precisely, we need an algorithm that, for given $n_1, \ldots, n_P, q_1, \ldots, q_P$, finds an x in N. We propose an ART-type method using functions

$$\alpha_k(x, t, z) = \begin{cases} x, & \text{if } \langle t, x \rangle \leq z, \\ x + \lambda^{(k)} \frac{z - \langle t, x \rangle}{\langle t, t \rangle} t, & \text{otherwise,} \end{cases} \tag{11.9}$$

where $\lambda^{(k)}$ is a real number, referred to as the *relaxation parameter.*

Consider the following procedure, which we refer to as the *relaxation method for inequalities.*

$$\begin{aligned} &x^{(0)} \text{ is arbitrary,} \\ &x^{(k+1)} = \alpha_k(x^{(k)}, n_{i_k}, q_{i_k}), \end{aligned} \tag{11.10}$$

where α_k is defined by (11.9) and $i_k = k(\mathrm{mod}\ P) + 1$. If the relaxation parameters satisfy the weak condition that, for some ε_1 and ε_2 and for all k,

$$0 < \varepsilon_1 \le \lambda^{(k)} \le \varepsilon_2 < 2, \tag{11.11}$$

then the relaxation method for inequalities produces a sequence $x^{(0)}, x^{(1)}, x^{(2)}$, ... that converges to a vector in N, provided only that N is not empty. Proof of this result appears in Section 15.8.

Next we discuss the geometrical nature of the relaxation method for inequalities and, in particular, the role of the relaxation parameters. Let

$$H_i = \{x \mid \langle n_i, x\rangle = q_i\}\,. \tag{11.12}$$

In words, H_i is the set of vectors x for which the ith inequality is satisfied by the two sides of the inequality being actually equal. Note that H_i is a subset of N_i. Each H_i is what mathematicians call a *hyperplane*. If the dimension J of x is three, then a hyperplane is a plane in three-dimensional space. If the dimension J of x is two, then a hyperplane is a straight line in two-dimensional space (i.e., in a plane).

The two-dimensional case is illustrated in Fig. 11.1. There are two hyperplanes H_1 and H_2. For these hyperplanes

$$n_1 = \begin{pmatrix} 4 \\ 1 \end{pmatrix}, \quad n_2 = \begin{pmatrix} 2 \\ 5 \end{pmatrix}, \tag{11.13}$$

$$q_1 = 24, \qquad q_2 = 30. \tag{11.14}$$

Observe the following simple geometrical fact: The vector n_i (think of it as the line drawn from the origin to the point n_i in the plane) is perpendicular to the line H_i. This statement generalizes to any dimensions, as a reader acquainted with analytic geometry can easily see from (11.12).

Since in the relaxation method for inequalities, (11.10), the role of t in (11.9) is taken by n_{i_k}, we see that if $x^{(k+1)}$ differs from $x^{(k)}$ at all, then $x^{(k+1)} - x^{(k)}$ (which is a multiple of n_{i_k}) is perpendicular to H_{i_k}.

The geometrical interpretation in the two-dimensional case is that N_i is a set of points lying on one side of the line H_i (including the line H_i). For example, in Fig. 11.1, both N_1 and N_2 are those half-planes that include the origin. Their intersection N, is shown dark in Fig. 11.1. This notion also generalizes, and we say that N_i is a *half-space* that is the set of points lying on one side of the hyperplane H_i (including the hyperplane H_i). The set N, as defined by (11.8), is an intersection of such half-spaces.

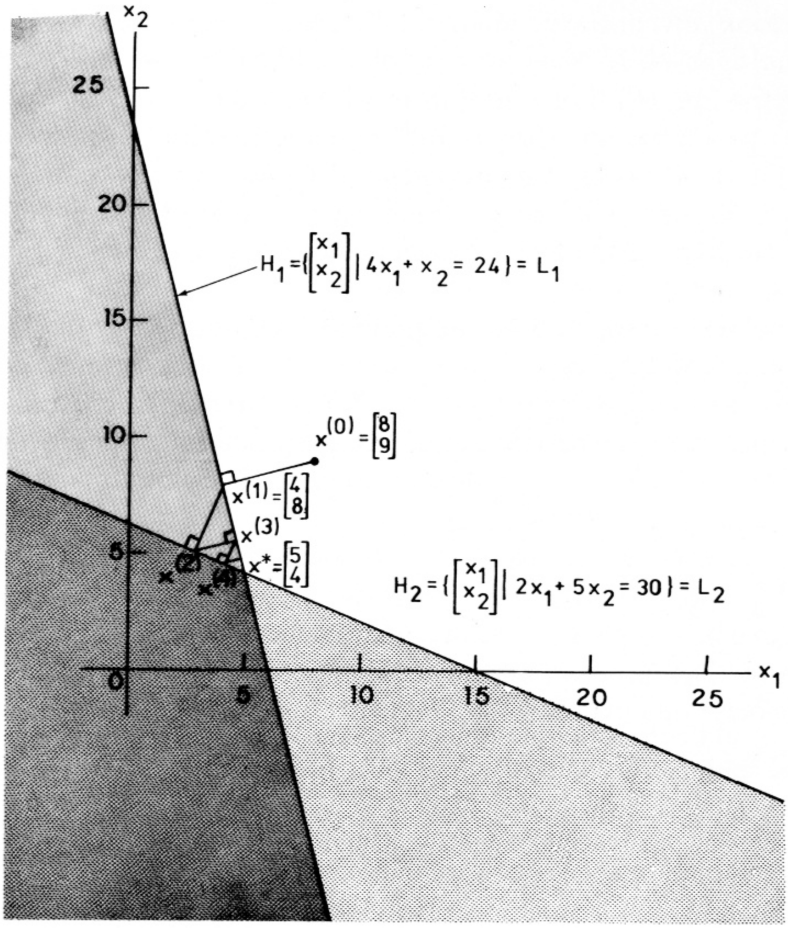


Fig. 11.1: Demonstration of the relaxation method (with $\lambda^{(k)} = 1$, for all k) for the simple case when $I = J = 2$. In the demonstration n_1 and n_2 are given by (11.13) and q_1 and q_2 by (11.14). (Illustration based on [127], Copyright 1976, with permission from Elsevier.)

In the relaxation method for inequalities, see (11.9) and (11.10), if $x^{(k)}$ is in the half-space N_{i_k} then we do not change our estimate during the kth iterative step (i.e., $x^{(k+1)} = x^{(k)}$). If $x^{(k)}$ is not in the half-space N_{i_k}, then we move our estimate perpendicular to the bounding hyperplane H_{i_k} of N_{i_k} (i.e., $x^{(k+1)} - x^{(k)}$ is orthogonal to H_{i_k}). The amount of movement depends on the size of $\lambda^{(k)}$. If $\lambda^{(k)} = 1$, then by an argument identical to that given for (11.3), we can prove that $\langle n_{i_k}, x^{(k+1)} \rangle = q_{i_k}$, i.e., that $x^{(k+1)}$ is in the hyperplane H_{i_k}. The following statement can be shown in a similarly easy fashion. During the kth step of the relaxation method for inequalities with $x^{(k)}$ not in N_{i_k}, the move from $x^{(k)}$ to $x^{(k+1)}$ is perpendicular to H_{i_k} and has one of the following geometrical properties:

if $\lambda^{(k)} < 0$, the move is away from H_{i_k};
if $\lambda^{(k)} = 0$, there is no movement;
if $0 < \lambda^{(k)} < 1$, the move is toward H_{i_k}, but does not quite reach it;
if $\lambda^{(k)} = 1$, the move is to H_{i_k} exactly;
if $1 < \lambda^{(k)} < 2$, the move is past H_{i_k}, but $x^{(k+1)}$ is nearer to H_{i_k};
if $\lambda^{(k)} = 2$, $x^{(k+1)}$ is the mirror image (reflection) of $x^{(k)}$ in H_{i_k};
if $\lambda^{(k)} > 2$, $x^{(k+1)}$ is on the other side of H_{i_k} further from H_{i_k} than $x^{(k)}$.

To illustrate the relaxation method for inequalities, consider again Fig. 11.1. Two inequalities are involved (i.e., $P = 2$), and the vectors n_1, n_2 and the scalars q_1, q_2 are defined by (11.13) and (11.14), respectively. Suppose we choose $\lambda^{(k)} = 1$, for all k. We also have to choose the initial vector. If we let

$$x^{(0)} = \begin{pmatrix} 8 \\ 9 \end{pmatrix}, \tag{11.15}$$

then, as can easily be checked,

$$x^{(1)} = \begin{pmatrix} 4 \\ 8 \end{pmatrix} \quad \text{and} \quad x^{(2)} = \begin{pmatrix} 80/29 \\ 142/29 \end{pmatrix}. \tag{11.16}$$

Since $x^{(2)}$ is in both N_1 and N_2, all values of $x^{(k)}$, for $k \geq 2$, are the same as $x^{(2)}$. Hence the method converges to $x^* = x^{(2)}$, which is in N.

The convergence that occurs in this example is called *finite convergence* since the $x^{(k)}$ remain constant after a finite number of iterative steps. It is *not* the case that the relaxation method for inequalities always has finite convergence if the $\lambda^{(k)}$ are chosen according to (11.11).

Now we turn to study systems of equations. Suppose we are given J-dimensional vectors a_i and real numbers b_i for $1 \leq i \leq I$. Let,

$$L_i = \{x \mid \langle a_i, x\rangle = b_i\}, \tag{11.17}$$

and

$$L = \bigcap_{i=1}^{I} L_i. \tag{11.18}$$

We observe the following fact: L can also be expressed as the intersection of a set of half-spaces. In fact, if we let $P = 2I$ and define, for $1 \leq i \leq I$,

$$N_{2i-1} = \{x \mid \langle -a_i, x\rangle \leq -b_i\}, \tag{11.19}$$

$$N_{2i} = \{x \mid \langle a_i, x\rangle \leq b_i\}, \tag{11.20}$$

then

$$L_i = N_{2i-1} \cap N_{2i} \tag{11.21}$$

and, consequently, L defined by (11.18) is the same as N defined by (11.8), provided that the N_i are defined by (11.19) and (11.20). Hence we can apply the relaxation method for a system of inequalities to find an element of L. Here we assume that, for all $1 \le i \le I$, $\langle a_i, a_i \rangle > 0$.

However, note that

$$L_i = H_{2i-1} = H_{2i}, \tag{11.22}$$

where H_i denotes the bounding hyperplane of the half-space N_i, see (11.12). Hence the combined effect of the $(2k-1)$st and $(2k)$th iterative steps is to move (if at all) perpendicular to the hyperplane L_{i_k}. We can combine these two steps and obtain the following *relaxation method for systems of equalities:*

$$\begin{aligned} &x^{(0)} \text{ is arbitrary}, \\ &x^{(k+1)} = x^{(k)} + c^{(k)} a_{i_k}. \end{aligned} \tag{11.23}$$

It follows easily from the result stated for the convergence of the relaxation method for inequalities that if, for all $k \ge 0$,

$$c^{(k)} = \lambda^{(k)} \frac{b_{i_k} - \langle a_{i_k} x^{(k)} \rangle}{\langle a_{i_k}, a_{i_k} \rangle}, \tag{11.24}$$

with $\lambda^{(k)}$ satisfying (11.11), then the relaxation method for equalities produces a sequence $x^{(0)}, x^{(1)}, x^{(2)}, \dots$ that converges to a vector in L, provided only that L is not empty.

The algorithm described by (11.1) and (11.2) is a special case of the relaxation method for equalities, with $\lambda^{(k)} = 1$, for all k. We illustrate this special case in Fig. 11.1. Defining $a_1 = n_1$, $a_2 = n_2$, $b_1 = q_1$ and $b_2 = q_2$ by (11.13) and (11.14), we see that if we start with $x^{(0)}$ defined by (11.15), then we get $x^{(1)}$ and $x^{(2)}$ as defined by (11.16). From this point on, the sequence produced by the relaxation method for equalities differs from the relaxation method for inequalities, which we discussed before. This is because $x^{(2)}$ does not lie in L_2 and so further steps are necessary to get nearer and nearer to an element of L, which, in this case, is the unique intersection x^* of the two lines L_1 and L_2. The geometrical interpretation given above shows that the sequence of vectors is produced by dropping perpendiculars alternately onto L_1 and L_2, see Fig. 11.1.

In general, L has more than one element. With a little extra care in choosing $x^{(0)}$, we can ensure that the relaxation method for equalities converges to an element of L that satisfies an optimization criterion, namely, the minimum norm criterion; see (6.37) in Section 6.4. To do this, we introduce a set S of vectors, which is the set of all linear combinations of the a_i, that is

$$S = \left\{ x \mid x = \sum_{i=1}^{I} \beta_i a_i \text{ for some real numbers } \beta_i \right\}. \tag{11.25}$$

The following result is sometimes referred to as the *minimum norm theorem*: if L is not empty, then there exists one, and only one, element x^* in $L \cap S$; furthermore, for all x in L other than x^*,

$$\|x^*\| < \|x\|, \tag{11.26}$$

where $\|x\|^2 = \langle x, x\rangle$; see (6.37). We refer to x^* as the *minimum norm element* of L. We have already discussed in Section 6.4 why choosing a minimum norm element may be considered useful in picture reconstruction.

It is clear from (11.23) that if we choose $x^{(0)}$ to be an element of S, then $x^{(k)}$ is an element of S, for all k. It follows, using basic linear algebra, that the limit x^* of the sequence $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$, is also in S. Since the limit x^* is also in L (by the convergence of the relaxation method for equalities), x^* is in $L \cap S$. Hence, by the minimum norm theorem, x^* is the minimum norm element of L.

In the example for Fig. 11.1, S is the set of all two-dimensional vectors. Hence $x^{(0)}$ is in S, whichever way it is chosen. Since in that simple example there is only one solution, it is by necessity the minimum norm solution. This is, however, not typical. If there are more unknowns than equations, there will be invariably many solutions, and care has to be taken in choosing $x^{(0)}$ if the minimum norm solution is desired.

There are versions of the relaxation method that provide the minimum norm solution for a system of inequalities. These are more complex than the methods previously discussed and are beyond the scope of this book. In the next section we discuss a method whose implementation is similar to the implementation of relaxation methods for finding the minimum norm solution for a system of inequalities.

11.3 Additive ART

In this section we discuss the application of the relaxation methods of the last section to image reconstruction. Such methods are referred to as *additive* ART, since in a single iterative step the current iterate is altered by adding to it a scalar multiple of the transpose of a row of the projection matrix; see (11.1) and (11.2).

The simplest approach is to use the relaxation method for equalities, described by (11.23) and (11.24), with $a_i = r_i$ and $b_i = y_i$. By the result stated in the previous section, this generates a sequence of vectors $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ that converges to an x^* such that $Rx^* = y$, provided that there is such an x^*. The problem is that the relationship between the image vector x and the measurement vector y (6.24) is such that there may not exist such an x^* or that, even if such an x^* exists, it may not be a desirable solution to the discrete reconstruction problem. In view of this, it is pleasantly surprising that even this simple approach leads to acceptable reconstructions, especially if the relaxation parameters are chosen to be rather small (e.g., 0.05). This is illustrated in Section 11.5.

One way of making the theory of the last section applicable to the image reconstruction problem is by using the formulation that involves a system of inequalities (6.42). Much work has been done in that direction. In particular, ART-type procedures exist that find the minimum norm solution of a system of inequalities. Such procedures require a formulation more complicated than what is provided by the framework of (11.1). In addition to the sequence of J-dimensional vectors $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$, they produce and use a sequence of I-dimensional vectors $u^{(0)}, u^{(1)}, u^{(2)}, \ldots$.

In this book we discuss an alternative approach, but one that has a similar implementation. We give an additive ART method for finding the Bayesian estimate (see Section 6.4) under certain restrictive assumptions.

In the terminology of Section 6.4, we make the following assumptions. Both X and E are multivariate Gaussian random variables, with μ_E the zero vector, and V_X and V_E both multiples of identity matrices of appropriate sizes. In other words, we assume that components of a sample of $X - \mu_X$ are uncorrelated, and that each component is a sample from the same Gaussian random variable; and we also assume that components of a sample of E are uncorrelated and that each component is a sample from the same zero mean Gaussian random variable.

We use t^2 to denote the diagonal entries of V_X and s^2 to denote the diagonal entries of V_E and let $r = t/s$. It follows from the discussion around (6.31) that the dimensionality of t, and hence of r, is inverse length. According to (6.33), the Bayesian estimate is the vector x that minimizes

$$r^2 \left\| y - Rx \right\|^2 + \left\| x - \mu_X \right\|^2 . \tag{11.27}$$

Note that a small value of r indicates that prior knowledge of the expected value of the image vector is important relative to the measured data, while a large value of r indicates the opposite.

What we are going to do now is essentially the following. We look at the equation $Rx + e = y$ as an equation in $I + J$ unknowns, namely all the components of x and all the components of e. This is a consistent system of equations; for any x, $e = y - Rx$ provides a solution. Methods for solving consistent systems can therefore be applied. However, in order to find the x that minimizes (11.27) a slightly more complicated approach is needed.

We denote column vectors of dimension $I + J$ by

$$\begin{pmatrix} u \\ z \end{pmatrix},$$

where u has I components and z has J components. We also use the notation

$$(U \quad rR)$$

for the $I \times (I + J)$ matrix, whose first I columns form the $I \times I$ identity matrix U and whose last J columns form the matrix R with every entry multiplied by r. The system of equations

$$(U \quad rR) \begin{pmatrix} u \\ z \end{pmatrix} = r \left(y - R\mu_X \right) \tag{11.28}$$

is a consistent system of equations. This is because, if we let $\hat{z}$ be an arbitrary J-dimensional vector and let

$$\hat{u} = r(y - R\mu_X - R\hat{z}), \tag{11.29}$$

then $\begin{pmatrix} \hat{u} \\ \hat{z} \end{pmatrix}$ satisfies (11.28).

The reason for introducing (11.28) is the following. If u^* and z^* are vectors such that $\begin{pmatrix} u^* \\ z^* \end{pmatrix}$ is the minimum norm solution of (11.28) and if

$$x^* = z^* + \mu_X, \tag{11.30}$$

then x^* minimizes (11.27).

In order to verify this claim consider any J-dimensional vector $\hat{x}$. Let

$$\hat{z} = \hat{x} - \mu_X \tag{11.31}$$

and define $\hat{u}$ by (11.29). Then

$$\hat{u} = r(y - R\hat{x}). \tag{11.32}$$

It follows that

$$r^2 \|y - R\hat{x}\|^2 + \|\hat{x} - \mu_X\|^2 = \|\hat{u}\|^2 + \|\hat{z}\|^2 \geq \|u^*\|^2 + \|z^*\|^2, \tag{11.33}$$

since $\begin{pmatrix} u^* \\ z^* \end{pmatrix}$ is the minimum norm solution of (11.28) and $\begin{pmatrix} \hat{u} \\ \hat{z} \end{pmatrix}$ is also a solution of (11.28).

From the fact that u^*, z^*, and x^* satisfy (11.28) and (11.30), we obtain

$$u^* = r\left(y - Rx^*\right). \tag{11.34}$$

This combined with (11.30) and (11.33) gives

$$r^2 \|y - R\hat{x}\|^2 + \|\hat{x} - \mu_X\|^2 \geq r^2 \|y - Rx^*\|^2 + \|x^* - \mu_X\|^2. \tag{11.35}$$

Since $\hat{x}$ is an arbitrary J-dimensional vector, this shows that x^* minimizes (11.27).

It follows therefore that any method that provides the minimum norm solution of (11.28) automatically gives us the vector that minimizes (11.27). One way of finding the minimum norm solution of a consistent system of equalities is the relaxation method for equalities. Note that the iterative step of (11.23) applied to (11.28) is

$$\begin{pmatrix} u^{(k+1)} \\ z^{(k+1)} \end{pmatrix} = \begin{pmatrix} u^{(k)} \\ z^{(k)} \end{pmatrix} + c^{(k)} \begin{pmatrix} e_{i_k} \\ r\, r_{i_k} \end{pmatrix}, \tag{11.36}$$

where e_i denotes the transpose of the ith row of E (which happens to be the same as the ith column of E, since E is an identity matrix), and

$$c^{(k)} = \lambda^{(k)} \frac{r\left(y_{i_k} - \langle r_{i_k}, \mu_X\rangle\right) - \left(u_{i_k}^{(k)} + r\left\langle r_{i_k}, z^{(k)}\right\rangle\right)}{1 + r^2 \left\|r_{i_k}\right\|^2}. \tag{11.37}$$

Note that, if S is defined by (11.25), then the zero vector is in S. Hence, one way of ensuring that the relaxation method for equalities, with iterative steps as in (11.36), converges to the minimum norm solution of (11.28) is to choose both $u^{(0)}$ and $z^{(0)}$ to be zero vectors of appropriate dimensions.

We define, for all k,

$$x^{(k)} = z^{(k)} + \mu_X. \tag{11.38}$$

If the sequence $z^{(0)}, z^{(1)}, z^{(2)}, \ldots$ converges to z^*, then the sequence $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ converges to x^*, defined by (11.30). This x^* minimizes (11.27).

There is in fact no need to introduce explicitly the $z^{(k)}$ into the algorithm. Combining (11.36), (11.37), and (11.38) with the fact that both $u^{(0)}$ and $z^{(0)}$ are chosen to be zero vectors, we get the following algorithm. The sequence $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ produced by it converges to the Bayesian estimate x^*, provided that the relaxation parameters $\lambda^{(k)}$ satisfy (11.11).

$$\begin{aligned}
&u^{(0)} \text{ is the } I\text{-dimensional zero vector,}\\
&x^{(0)} = \mu_X,\\
&u^{(k+1)} = u^{(k)} + c^{(k)} e_{i_k},\\
&x^{(k+1)} = x^{(k)} + r c^{(k)} r_{i_k},
\end{aligned} \tag{11.39}$$

where

$$c^{(k)} = \lambda^{(k)} \frac{r\left(y_{i_k} - \left\langle r_{i_k}, x^{(k)}\right\rangle\right) - u_{i_k}^{(k)}}{1 + r^2 \left\|r_{i_k}\right\|^2}. \tag{11.40}$$

Note that this algorithm cannot be brought into the framework of (11.1), but its implementation is hardly more complicated than the implementation of the method described by (11.2). We need an additional sequence of I-dimensional vectors $u^{(k)}$, but in the kth iterative step, only one component of $u^{(k)}$ (namely the i_kth component) is needed or altered. As pointed out in Section 11.1, in our application area the r_{i_k} are usually not stored at all, but the location and size of their nonzero elements are calculated as and when needed. Hence the algorithm described by (11.39) and (11.40) shares the storage-efficient nature of the simple ART method described in Section 11.1. It is easy to see that the computational requirements are also essentially the same.

The algorithm described by (11.39) and (11.40) is a typical additive ART algorithm. To illustrate this further, we now state, without proof, an additive ART algorithm that produces a sequence $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ that converges to the minimum norm solution of a system of two-sided inequalities

$$\gamma_i \leq \langle r_i, x \rangle \leq \delta_i, \tag{11.41}$$

$1 \leq i \leq I$; compare this with (6.40).

$$\begin{aligned} &u^{(0)} \text{ is the } I\text{-dimensional zero vector,} \\ &x^{(0)} \text{ is the } J\text{-dimensional zero vector,} \\ &u^{(k+1)} = u^{(k)} + c^{(k)} e_{i_k}, \\ &x^{(k+1)} = x^{(k)} + c^{(k)} r_{i_k}, \end{aligned} \tag{11.42}$$

where

$$c^{(k)} = \text{mid}\left\{u_{i_k}^{(k)}, \left(\delta_{i_k} - \left\langle r_{i_k}, x^{(k)} \right\rangle\right) / \|r_{i_k}\|^2, \left(\gamma_{i_k} - \left\langle r_{i_k}, x^{(k)} \right\rangle\right) / \|r_{i_k}\|^2\right\}, \tag{11.43}$$

where $\text{mid}\{u, v, w\}$ denotes the median of the three real numbers u, v, and w. The algorithm described by (11.42) and (11.43) has been referred to as ART4 in the literature, in order to distinguish it from other versions of ART, which have different convergence properties.

11.4 Tricks

It has been found in practice that the efficiency of iterative algorithms for image reconstruction can often be improved by applying between iterative steps certain processes to the image vectors. These processes have been referred to as *tricks* in the literature. In this section we deal with such tricks and also with other recommendations that can be made based on experience to improve the performance of ART, especially the potential usefulness of the images obtained by the early iterations of the algorithm.

Consider the iterative step in ART as described by (11.1). Let τ_k be functions mapping J-dimensional vectors into J-dimensional vectors. Then the iterative method of (11.1) combined with the sequence of tricks τ_k produces a sequence $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ defined by

$$\hat{x}^{(k+1)} = \alpha_k \left(x^{(k)}, r_{i_k}, y_{i_k}\right), \tag{11.44}$$

$$x^{(k+1)} = \tau_{k+1}\left(\hat{x}^{(k+1)}\right). \tag{11.45}$$

Tricks are useful if they incorporate prior knowledge about the desirable image vectors. Sometimes they can be used to accelerate convergence towards the image vector that satisfies the specified optimization criterion. Other times, they actually cause the process to move towards an image vector other than the one optimizing the specified criterion function, but that is nevertheless a better approximation of the picture to be reconstructed according to some evaluation criterion such as the picture distance measures of Section 5.1. The latter happens, for instance, if the desirable digitized pictures have a common property that cannot be expressed by a simple function, but that may be obtained by the application of an appropriate trick. In the discussions that follow, the intuitive justifications for all the tricks are based on the assumption that the basis functions are those associated with an $n \times n$ digitization, defined by (6.17).

One possible trick is based on the idea of *selective smoothing*, discussed in Section 5.3. It is potentially useful when the image to be reconstructed is made up from regions within which the values are largely uniform and distinguishable from those in other regions. (Figures 5.3 and 5.4 illustrate the effect of a single application of the trick of selective smoothing to the output of FBP for divergent beams. Table 5.1 shows that improvements in the picture distance measures are achieved using this trick. There was also an improvement in the HITR in our task-oriented experiment, but not in the IROI.) When this trick is used in conjunction with ART, typically we choose τ_k in (11.45) to represent selective smoothing only infrequently, e.g., only when k is a multiple of I (the number of measurements). For other values of k, we choose τ_k to be the identity function, which does not change the image vector.

In contrast, the trick of constraining is usually applied at every iterative step of ART. *Constraining* is justified in case we have prior information about the range within which the components of acceptable image vectors must lie. For example, the linear attenuation coefficient (at any energy) is always nonnegative, and in medical applications we may usually assume that it is always bounded above by the linear attenuation coefficient of compact bone. Such constraints may be introduced into series expansion methods in various ways. They may simply be made part of the set of inequalities (11.6). Or they may be introduced into the iterative algorithm as tricks. For example, if we know that, for $1 \le j \le J$,

$$\lambda \le x_j \le \mu, \tag{11.46}$$

then the following trick is appropriate.

$$\tau_k(\hat{x}) = x, \tag{11.47}$$

where, for $1 \le j \le J$,

$$x_j = \begin{cases} \lambda, & \text{if} \quad \hat{x}_j < \lambda, \\ \hat{x}_j & \text{if} \quad \lambda \le \hat{x}_j \le \mu, \\ \mu, & \text{if} \quad \mu < \hat{x}_j. \end{cases} \tag{11.48}$$

Such a trick can be easily incorporated into ART. To demonstrate this, consider the relaxation method for inequalities. The following is claimed to be true. If N, as in (11.8), contains at least one vector x whose components satisfy (11.46), then the algorithm below produces a sequence $x^{(0)}, x^{(1)}, x^{(2)}, \ldots$ that converges to an element of N whose components satisfy (11.46).

$$\begin{aligned} x^{(0)} & \quad \text{is arbitrary}, \\ \hat{x}^{(k+1)} &= \alpha_k \left(x^{(k)}, n_{i_k}, q_{i_k}\right), \\ x^{(k+1)} &= \tau_{k+1}\left(\hat{x}^{(k+1)}\right), \end{aligned} \tag{11.49}$$

where α_k is defined by (11.9) with the λ_k satisfying (11.11), and τ_k is defined by (11.47) and (11.48).

The verification of this claim follows easily from the convergence of the relaxation method for inequalities. This is because the set of vectors M that satisfy (11.46) can be characterized as follows. Let, for $1 \leq j \leq J$,

$$M_{2j-1} = \{x \mid x_j \leq \mu\} \tag{11.50}$$

and

$$M_{2j} = \{x \mid -x_j \leq -\lambda\}. \tag{11.51}$$

Then

$$M = \bigcap_{j=1}^{2J} M_j. \tag{11.52}$$

Thus M, the set of vectors satisfying (11.46), can be described in a way strictly analogous to the way N is described in Section 11.2. The reader can easily check that applying the relaxation method for inequalities based on M instead of N, with relaxation parameter 1 and initial vector $\hat{x}$ produces in $2J$ iterative steps the vector $\tau_k(\hat{x})$ where τ_k is defined by (11.47) and (11.48). Thus, the trick of constraining in this case is equivalent to applying the relaxation method to a larger set of inequalities. This completes the verification of the claim on the convergence of (11.49).

There are other versions of constraining in use besides the one specified by (11.47) and (11.48). For example, there is a way of defining the constraining τ_ks so that, when used in conjunction with the algorithm described in (11.42), the method converges to the minimum norm solution of the combined system (11.41) and (11.46). Another method, which is useful when it is known a priori that there are only two different densities in the picture (as is the case in certain nondestructive testing applications), is to use τ_ks that set the values of $\hat{x}_j$ to either one or the other of the two densities.

Another trick that we have already come across is *normalization*. This is discussed in conjunction with the backprojection method in Section 7.2. Repeated normalization during the iterative procedure has sometimes been found to improve the speed of convergence of ART to a desirable result.

A trick whose use is relatively recent is referred to as *superiorization*. The idea is the following. Suppose that we have a secondary optimization criterion of the kind discussed in Section 6.4 for which we do not have an efficient algorithm that converges to the optimal solution for a given set of equality and/or inequality constraints; a possible example is total variation (6.45) minimization. Then we can select the τ_ks so that they steer the iterative process in the direction of the optimal solution. If the criterion is to find an x

satisfying the constraints for which $\phi(x)$ is small for some functional ϕ (such as provided by TV), then we can use

$$\tau_k(x) = x - \beta_k \nabla\phi(x), \tag{11.53}$$

where the vector $\nabla\phi(x)$ (called the *gradient* of ϕ at x) is the direction of greatest increase in ϕ at the vector x and the β_k are positive real numbers. For appropriate choices of the β_k it can be proved that if an ART procedure converges to a vector satisfying a set of equality/inequality constraints, then the same procedure altered by the trick of (11.53) also converges to such a vector. The expectation (validated by experience) is that for the procedure with the trick we get to an x for which $\phi(x)$ is smaller than it would be without the trick. Ideally, we would like to get to an x for which $\phi(x)$ is as small as possible, but this is not guaranteed; this is why the trick is referred to as superiorization (as opposed to optimization). In practice the trick of superiorization, just like the trick of selective smoothing, is applied only infrequently during the iterative process.

Although there are other tricks whose use has been reported in the literature, we conclude this section by a discussion of four topics related to, but somewhat different from, tricks.

An essential tool available with ART is the relaxation parameter. We have already mentioned that choosing a low value for the relaxation parameter has been found to result in good reconstructions using ART-type algorithms, even on experimentally obtained data. A low relaxation parameter, *underrelaxation*, seems to reduce the effect of inaccuracies in the equations, and prevents the noisy appearance of ART-type reconstructions when using a high relaxation parameter. This is illustrated in the next section.

In certain situations a limited use of a high relaxation parameter is advisable. When solving a system of inequalities, the process can be markedly shortened if, whenever an inequality is only slightly violated by the current iterate, a relaxation parameter with value 2 is used, resulting in a mirror *reflection* of the iterate in the bounding hyperplane associated with the inequality. Selective use of reflections can, under certain circumstances, ensure finite convergence.

The choice of *initialization* (i.e., of $x^{(0)}$) has an effect on the outcome of the iterative procedure, especially since due to time and cost constraints the number of iterative steps may be rather limited. For example, in the algorithm (11.39) $x^{(0)}$ is supposed to be μ_X. In practice, it may be very difficult to find the mean of the multivariate random variable that represents the actual situation. Instead, outputs of other methods (such as FBP) have often been used as $x^{(0)}$ for ART. Even more frequent in practice is the use of a uniformly gray picture, possibly with the estimated average density in every pixel, which is what was done in all the ART experiments reported in the next section.

Last, but not least, the order of equations (or inequalities) in the system (the data access ordering discussed in Section 11.1) can also have a significant effect on the practical performance of the algorithm, especially on the early iterates. With data collection such as our standard geometry depicted in Fig. 5.5, it is tempting to use the *sequential ordering*: access the data in the order $g(-N\lambda, 0), g((-N+1)\lambda, 0), \ldots, g(N\lambda, 0), g(-N\lambda, \Delta), g((-N+1)\lambda, \Delta), \ldots, g(N\lambda, \Delta), \ldots, \ldots, g(-N\lambda, (M-1)\Delta), g((-N+1)\lambda, (M-1)\Delta), \ldots, g(N\lambda, (M-1)\Delta)$, where $g(\sigma, \beta)$ denotes here the measured value of what is mathematically defined in (10.2). However, this sequential ordering is inferior to what is referred to as the *efficient ordering* in which the order of projection directions $m\Delta$ and, for each view, the order of lines within the view is chosen so as to minimize the number of commonly intersected pixels by a line and the lines selected recently. This can be made mathematically precise by considering the decomposition into a product of prime numbers of M and of $2N+1$. SNARK09 calculates the efficient order, but the user needs to ensure that both M and of $2N+1$ decompose into several prime numbers, as is the case for our standard geometry for which $M = 720 = 2 \times 2 \times 2 \times 2 \times 3 \times 3 \times 5$ and $2N+1 = 345 = 3 \times 5 \times 23$. While the sequential ordering produces the sequences $m = 0, 1, 2, 3, 4, \ldots$ and $n = 0, 1, 2, 3, 4, \ldots$, the efficient ordering produces the sequences $m = 0, 360, 180, 540, 90, \ldots$ and $n = 0, 115, 230, 23, 138, \ldots$ These changes in data access ordering translate into faster initial convergence of ART, as is illustrated in Fig. 11.2 by plotting

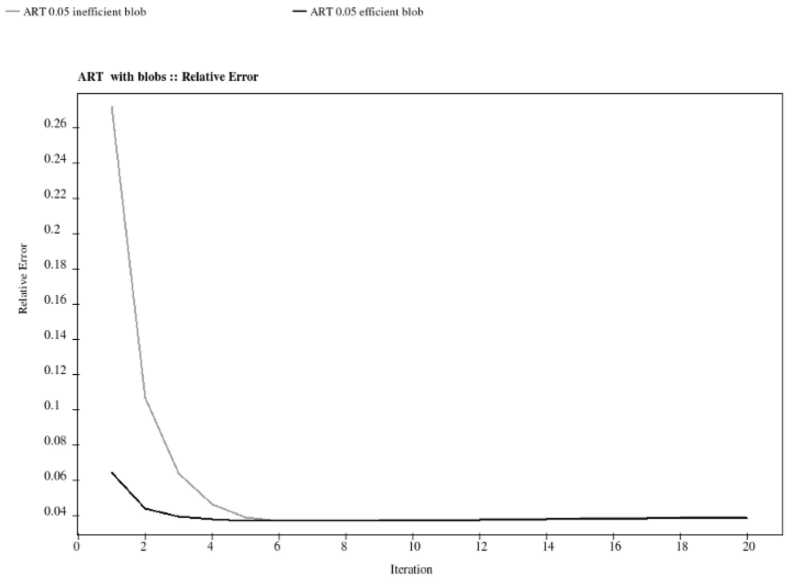


Fig. 11.2: Values of the picture distance measure r for ART reconstructions from the standard projection data with sequential ordering (light) and efficient ordering (dark), plotted at multiples of I iterations (complete cycles through the data).

the picture distance measure r against the number of times the algorithm cycled through all the data (all I equations). While it is clearly demonstrated that initially r gets reduced much faster with the efficient ordering, for this particular data set it does not seem to matter much, since both orderings need about five cycles through the data to obtain a near-minimal value of r. In other applications in which the number of projection directions is much larger (for example, in the order of 10,000 as is often the case in electron microscopy), one cycle through the data using the efficient ordering yields about as good a reconstruction as one is likely to get, but the sequential ordering needs several cycles through the data. In addition, as we demonstrate in the next section, the efficacy of the reconstruction produced by the efficient ordering may very well be superior to that produced by the sequential ordering.

11.5 Efficacy of ART

In this section we illustrate some of the algebraic reconstruction techniques. All the illustrations are done on the standard projection data. Only outputs at the end of some integer multiple of I iterations are used. This is because in I iterations the measurements for all source–detector positions have been made use of exactly once; i.e., we have cycled through the data exactly once. In all ART reconstructions reported in this section we initialized the process so that all components of x^0 are given the value of the estimated average density $\bar{x}$ based on the projection data, as specified in Section 6.4. We note that if we gave to the components of x^0 the value 0, the resulting $x^{(I)}$ would be indistinguishable from the $x^{(I)}$ we get by our selected initialization (with blobs, all $\lambda^{(k)} = 0.05$, efficient ordering, and no nonnegativity constraints).

We wish to emphasize first the importance of the basis functions. In Fig. 11.3 we plot the picture distance measure r against the number of times ART cycled through all the data, where we made the choices that the relaxation parameter is always 0.05 and the data access ordering is the efficient one. The two cases that we compare are when the basis functions are based on pixels

Table 11.1: Picture distance measures and timings (in seconds) for the reconstructions in Fig. 11.4. The last two columns report on the values of IROI and HITR for the various algorithms that were produced by a task-oriented evaluation experiment.

reconstruction in	d	r	t	IROI	HITR
Fig. 11.4(a)	0.1060	0.0423	8.7	0.1677	0.9499
Fig. 11.4(b)	0.0813	0.0327	29.4	0.1658	0.9213
Fig. 11.4(c)	0.0874	0.0470	29.2	0.1592	0.9198
Fig. 11.4(d)	0.0874	0.0373	163.7	0.1794	0.9481
Fig. 11.4(e)	0.0768	0.0488	66.2	0.1076	0.7128
Fig. 11.4(f)	0.0876	0.0391	148.9	0.1624	0.7820

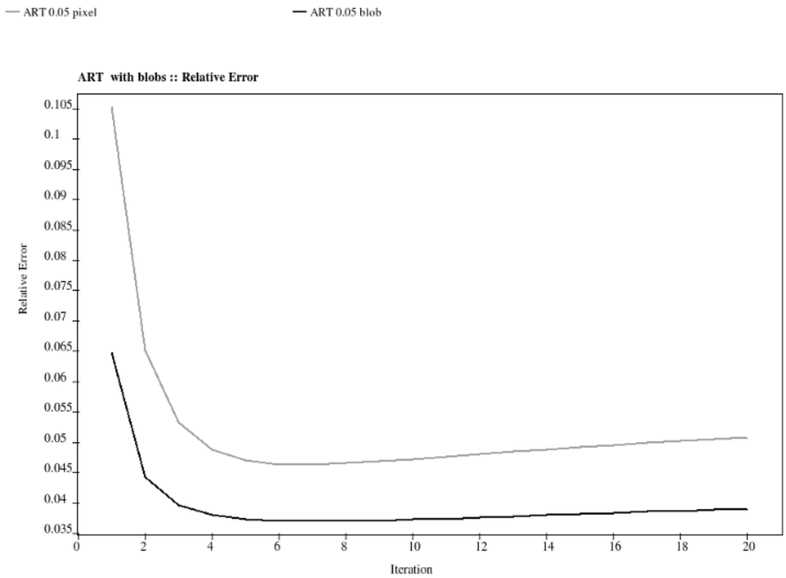


Fig. 11.3: Values of the picture distance measure r for ART reconstructions from the standard projection data with pixels (light) and blobs (dark), plotted at multiples of I iterations (complete cycles through the data).

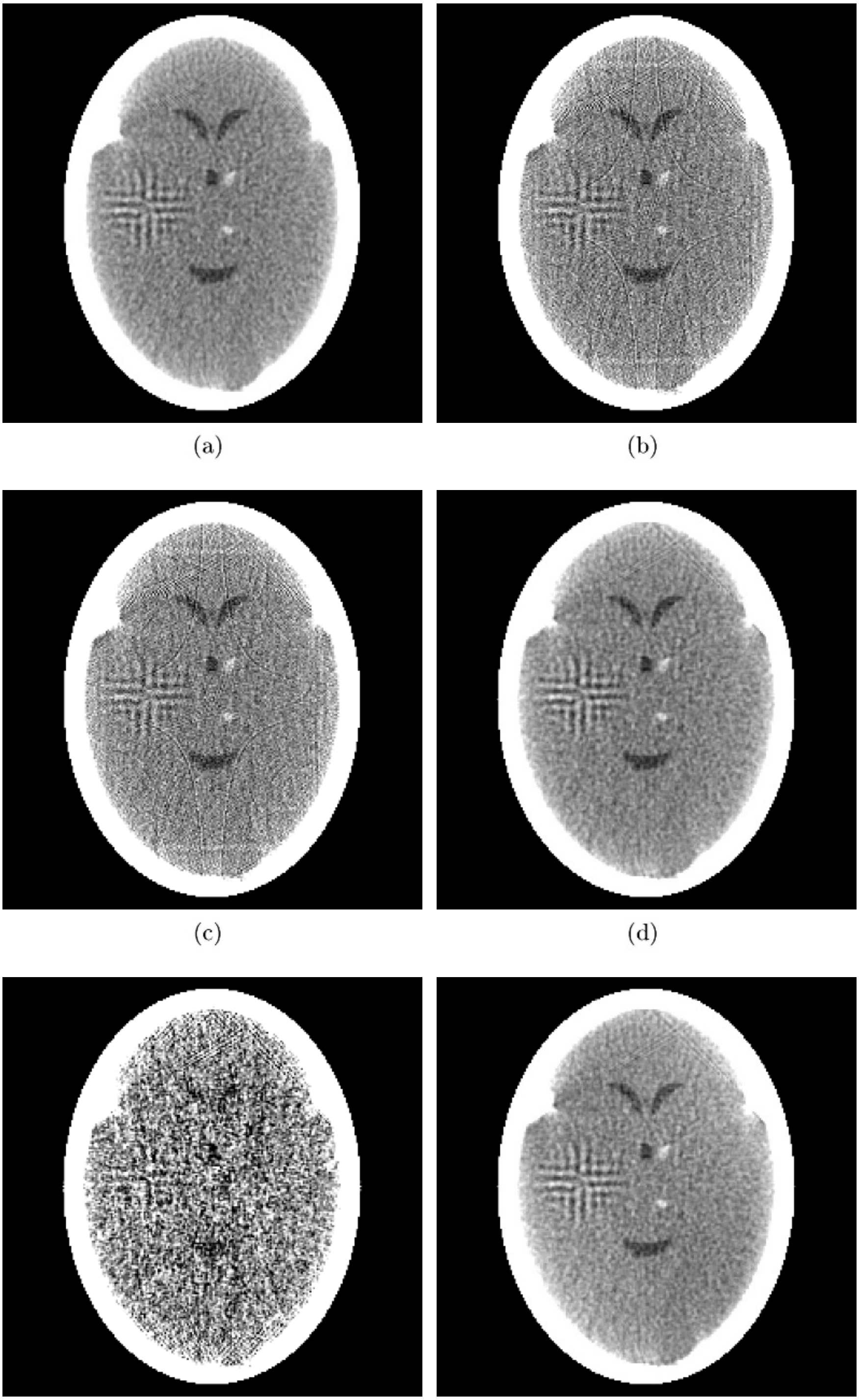
and when they are based on blobs as specified in Section 6.5. The results are quite impressive: as measured by r, blob basis functions are much better. The result of the $5I$th iteration of the blob reconstruction is shown in Fig. 11.4(d), while that of the $5I$th iteration of the pixel reconstruction is shown in Fig. 11.4(c). The blob reconstructions appears to be clearly superior. We attempted to improve the pixel reconstruction by enforcing nonnegativity on the reconstructed values, as can be done using ART by setting $\lambda = 0$ in (11.46). This does not result in a noticeable improvement in appearance, as can be seen in Fig. 11.4(b). By looking at Table 11.1, we see great improvements in the picture distance measures r and d as a result of enforcing nonnegativity, but this is totally misleading from the point of view of our application because the improvement is due to the values being more correctly reconstructed outside the skull (where the values in the phantom are all 0). From the points of view of the task-oriented figures of merit, IROI and HITR, ART with blobs is found superior to ART with pixels, with or without nonnegativity; the relevant P-values are all less than 10^{-10}. Plots of values along the 131st column of the ART with pixels and nonnegativity reconstruction and of the ART with blobs reconstruction are compared in Fig. 11.5.

Underrelaxation is also a must when ART is applied to real, and hence imperfect, data. In the experiments reported so far $\lambda^{(k)}$ was set equal to 0.05 for all k. If we do not use underrelaxation (that is we set $\lambda^{(k)}$ to 1 for all k), we get from the standard projection data the unacceptable reconstruction shown in Fig. 11.4(e). Note that in this case we used the $2I$th iterate, further iterations give worse results. The reason for this is in the nature of ART: after one iterative step with $\lambda^{(k)} = 1$, the associated measurement is satisfied exactly as shown in (11.5) and so the process jumps around satisfying the noise in the measurements. Underrelaxation reduces the influence of the noise. The correct value of the relaxation parameter is application dependent; the noisier the data the more we should be underrelaxing. Note in Table 11.1 that the figures of merit produced by the task-oriented studies without underrelaxation are much smaller than in all the other cases with which they are compared.

We now return to the issue of data access ordering. Using $5I$ iterations with $\lambda^{(k)} = 0.05$ for all k, we get from our standard projection data using the sequential ordering the reconstruction shown in Fig. 11.4(f), which does not look very different from the reconstruction obtained using the efficient ordering that is shown in Fig. 11.4(d). However, using either IROI or HITR as the figure of merit, results in our rejecting the null hypothesis that the two data access orderings are equally good in favor of the alternative hypothesis that the the efficient ordering is better with P-value less than 10^{-9}.

For comparison, we show in Fig. 11.4(a) the reconstruction from our standard projection data obtained by FBP for divergent beams with linear interpolation and sinc window (also called the Shepp–Logan window). The visual quality is similar to the best among the ART reconstructions that are reported, which is shown in Fig. 11.4(d). According to the picture distance measures ART is superior to FBP, and the same is true according to IROI with extreme significance (the P-value is less than 10^{-13}). According to HITR, FBP appears to be superior to ART, but the result is not particularly significant (the P-value is 0.0400, which means that even if the null hypothesis that the two methods are equally good were correct, there would be a 1 in 25 chance of observing a difference greater than what we observed). This experiment confirms the reports in the literature that ART with blobs, underrelaxation and efficient ordering outperforms FBP.

Fig. 11.4: Reconstructions from the standard projection data using (a) FBP for divergent beams with linear interpolation and sinc window (also called the Shepp–Logan window), (b) ART with pixels, $\lambda^{(k)} = 0.05$, $5I$th iteration, efficient ordering and nonnegativity, (c) ART with pixels, $\lambda^{(k)} = 0.05$, $5I$th iteration, efficient ordering and no nonnegativity, (d) ART with blobs, $\lambda^{(k)} = 0.05$, $5I$th iteration, efficient ordering and no nonnegativity, (e) ART with blobs, $\lambda^{(k)} = 1.0$, $2I$th iteration, efficient ordering and no nonnegativity, (f) ART with blobs, $\lambda^{(k)} = 0.05$, $5I$th iteration, sequential ordering and no nonnegativity.



(a) (b) (c) (d) (e) (f)

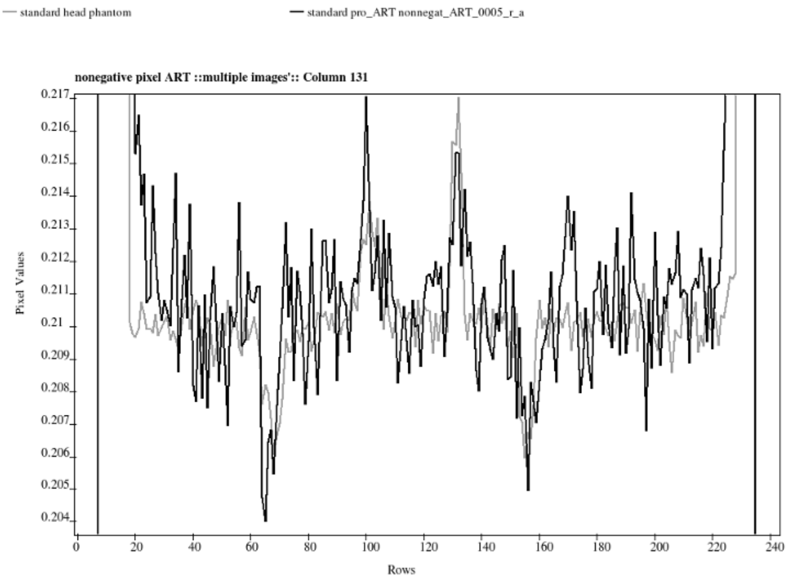
One thing is indisputable: the ART with blob reconstruction took nearly 19 times longer than FBP. However, this should not be the determining factor. The implementation of ART with blobs in SNARK09 is far from optimal and can be greatly improved. Also, computational speed keeps improving: the time reported in the first edition of this book (1980) for the ART with pixel reconstruction for a much smaller data set is 100 times longer than what we report here! A main advantage of ART over FBP is its flexibility. Even though in this section we have reported its application to data collected according to the standard geometry, the theory supports reconstruction from data collected over any set of lines. FBP-type algorithms need to be reinvented for each new mode of data collection, just as we had to do when we moved from parallel beams to divergent beams. Another aspect of the flexibility of ART is the ability to incorporate tricks and thereby steer the process towards a solution that is superior according to some criterion, such as TV.

In fact, ART can be used not only for superiorization but even for optimization of some fairly sophisticated functions. For example, the additive ART algorithm described in (11.39) and (11.40) converges to the Bayesian estimate x which minimizes (11.27). We delay the illustration of the usefulness of this until Section 13.3, where we demonstrate it on the reconstruction of a dynamic three-dimensional object, such as the heart.

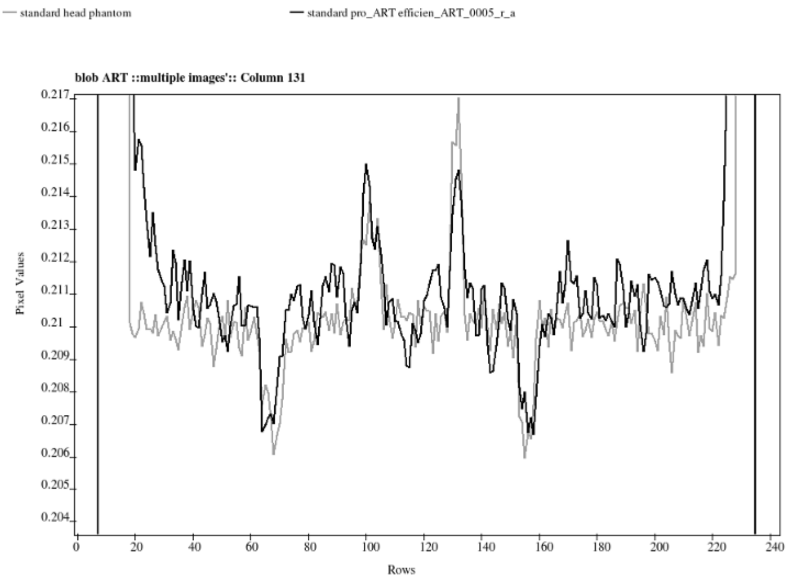
Notes and References

ART for image reconstruction was first introduced into the open literature in [99]. Coincidentally, essentially the same method had been already proposed for CT in a patent specification [145], originally filed in 1968. In fact, the simple procedure expressed in (11.2) was proposed in 1937 by S. Kaczmarz [154] for solving systems of consistent linear equations. An early tutorial on ART is [98], a more recent one is [117]. The methods discussed in this chapter are examples of the so-called row-action methods for solving very large sparse systems of equations and inequalities; for a survey, see [41, 46] and Chapter 6 of [48]. Such methods can be extended from finding common points of hyperplanes and half-spaces, as we discuss in this book, to finding common points of arbitrary closed convex sets (this is often referred to as POCS, short for projections onto convex sets); see, for example, Chapter 5 of [48] and [21]. An ART algorithm of this kind with a finite convergence property is ART3, described by [110]; for a faster version with an application to intensity modulated radiation therapy see [120].

Fig. 11.5: Plots of the head phantom of Fig. 4.6(a), shown light, and its reconstructions, shown dark, from the standard projection data using ART with $\lambda^{(k)} = 0.05$, efficient ordering and (a) pixels, 51th iteration and nonnegativity, and (b) blobs, 51th iteration and no nonnegativity.


standard head phantom
standard pro_ART nonnegat_ART_0005_r_a
nonegative pixel ART ::multiple images':: Column 131
Pixel Values
0.217
0.216
0.215
0.214
0.213
0.212
0.211
0.21
0.209
0.208
0.207
0.206
0.205
0.204
0
20
40
60
80
100
120
140
160
180
200
220
240
Rows

(a)


standard head phantom
standard pro_ART efficien_ART_0005_r_a
blob ART ::multiple images':: Column 131
Pixel Values
0.217
0.216
0.215
0.214
0.213
0.212
0.211
0.21
0.209
0.208
0.207
0.206
0.205
0.204
0
20
40
60
80
100
120
140
160
180
200
220
240
Rows

(b)

Our presentation of the relaxation method for solving systems of inequalities and equalities is based on [127] and [130], which reference earlier literature. The minimum norm theorem is a trivial consequence of what in optimization theory is referred to as the projection theorem (not to be confused with the theorem of the same name in image reconstruction); see, e.g., [190]. A relaxational approach to finding the minimum norm solution of systems of inequalities is presented in [177]. How such an approach translates into an algorithm for image reconstruction is discussed in [129].

Our discussion of the Bayesian approach is based on [123] and [124]; the former gives a detailed discussion of the validity of the assumptions made in the Bayesian approach. It can be generalized to minimize more complicated quadratic functions than the one in (11.27), see Appendix B of [185].

The expression "tricks" was first applied to the processes described in Section 11.4 in [127]. A treatment of the asymptotic behavior of iterative algorithms combined with the trick of selective smoothing is given in [64]. An algorithm that finds the minimum norm solution of the combined systems (11.41) and (11.46) is described in [129]; see also [220]. An algorithm for reconstructing objects with only two densities was given in [109]; the problem of reconstruction assuming only a few densities has developed into the field of discrete tomography [125]. The notion of superiorization, in particular in conjunction with total variation minimization, was introduced in [36]; see also [121]. A demonstration of the power of underrelaxation is given in [114]: it is shown that the trick of using a complimentary matrix (not discussed in this book) can be incorporated into ART as a special case of underrelaxation.

A theoretical study of the order in which views should be selected in ART was published in [106]. That paper, and its references, are also of interest, since they use a model for the image reconstruction problem different from anything discussed in this book. The outcome is an iterative procedure, which deals with pictures (as opposed to image vectors). Such procedures were referred to as continuous ART in [127]. The efficient ordering that we have proposed in Section 11.4 was first advocated in [135].

Some versions of ART that are not in this book are discussed in [98, 127]. Multiplicative ART (MART) is of particular interest, since it maximizes entropy (6.44), as proved in [176] and in [178]. MART is still the subject of active research and use [149, 162, 273, 275]. Another row-action optimization algorithm is RAMLA [34] that maximizes likelihood as defined in (6.47).

An alternative is block-ART that instead of treating single equations or inequalities one by one, tries to satisfy simultaneously a number of them (e.g., all that are associated with a single source position in the standard geometry). An early example of such an approach is [73]. The idea was rediscovered in [147], using the name ordered subsets, and became popular in emission tomography. Examples of recent papers on block-ART are [45, 47, 121].

For methods of accelerating algorithms of the type discussed above by using commonly available hardware see [23, 206, 274] and their references. For another recent implementational approach see [214].